Dear Mohammad Masudur,  
   
Thank you for your submission to the ICSE New Ideas and Emerging Research (NIER) track.    
   
We regret to inform you that your paper,  
   
Reusability Analysis of Crowdsource Code Examples  
   
has not been accepted for inclusion in the conference program.  We received over 146 submissions of which 35 were accepted, leading to an acceptance rate of 24%.  
   
At the end of this email you will find reviewers' comments for your submission, which we hope you will find useful for revising your paper.   
   
We have arranged with the workshop organizers to provide the opportunity for authors of rejected NIER or SEIP track papers to make late submissions to several of the workshops. As per standard submission policies, each paper may only be submitted to one workshop.    
   
The submission window for resubmitting your paper to a workshop is very limited.  Participating workshops will open their submission websites from 17th to 19th February only, so if you are interested in this opportunity you will have only a very short amount of time to choose a workshop and submit your paper.    
   
You may not submit new papers via this process, change authors, or change the title of your papers.   You may make minor modifications only.   
   
Participating workshops are listed here:  
   
\* (AST) 9th International Workshop on Automation of Software Test  
\* (CESI) 2nd International Workshop on Conducting Empirical Studies in Industry  
\* (CHASE) 7th International Workshop on Cooperative and Human Aspects of Software Engineering  
\* (CSI-SE) 1st International Workshop on CrowdSourcing in Software Engineering  
\* (CSTVA) 6th International Workshop on Constraints in Software Testing, Verification and Analysis  
\* (DAPSE) 2nd International Workshop on Data Analysis Patterns in Software Engineering  
\* (GREENS) 3rd International Workshop on Green and Sustainable Software  
\* (GTSE) 3rd SEMAT Workshop on General Theories of Software Engineering  
\* (MDSProPP) 1st International Workshop on Multi-Core/Many-Core Distributed Systems: Programming Paradigms to Practice  
\* (MiSE) 6th International Workshop on Modeling in Software Engineering  
\* (MoSEMInA) 1st International Workshop on Modern Software Engineering Methods for Industrial Automation  
\* (PESOS) 6th International Workshop on Principles of Engineering Service-Oriented and Cloud Systems  
\* (RAISE) 3rd International Workshop on Realizing Artificial Intelligence Synergies in Software Engineering  
\* (RCoSE) 1st International Workshop on Rapid Continuous Software Engineering  
\* (RSSE) 4th International Workshop on Recommendation Systems for Software Engineering  
\* (SBST) 7th International Workshop on Search-Based Software Testing  
\* (SE4SG) 3rd International Workshop on Software Engineering Challenges for the Smart Grid  
\* (SER&IPs) 1st International Workshop on Software Engineering Research and Industrial Practices  
\* (TwinPeaks) 4th International Workshop on the Twin Peaks of Requirements and Architecture  
\* (WETSoM) 5th International Workshop on Emerging Trends in Software Metrics  
If a workshop is not listed here then it is not participating in this late submission process.   
  
Please visit the ICSE Workshops website at   
[http://2014.icse-conferences.org/workshops](https://campus.usask.ca/owa/redir.aspx?C=yrRcAnczRk6IFuEE_LlaJxx8iTln_dAI5fk6PI8fnXc2A9D5ilUID1cNXLYdWgj0GwhZzbBlDBU.&URL=http%3a%2f%2f2014.icse-conferences.org%2fworkshops) to see additional information about each workshop and to access links to their submissions sites.  
   
As these papers are being submitted after the normal workshop deadlines,  workshop organizers will notify us of any submitted papers and we will forward the NIER reviews to them. If possible please add NIER [Paper Number] in the comments field when you submit your paper as this will help us respond to requests from the workshop chairs for reviews.  
   
Acceptance decisions at each workshop will be made independently by the workshop organizers.  
   
We hope to see you May 31-June 7, in Hyderabad, India for an exciting ICSE conference.  
   
Best Regards,  
   
Benoit Baudry and Jane Cleland-Huang  
NIER Co-Chairs  
  
  
----------------------- REVIEW 1 ---------------------  
PAPER: 153  
TITLE: Reusability Analysis of Crowdsource Code Examples  
AUTHORS: Mohammad Masudur Rahman, Chanchal Roy and Iman Keivanloo  
  
  
----------- REVIEW -----------  
The authors analyze code examples from StackOverflow to build a quality model. In the experiments the model agrees with the StackOverflow ratings in 74.54%.  
  
Building a quality model for code examples is an interesting idea. I like the metrics that the authors used in their model. The results are also promising.   
  
The authors try to accomplish a lot within 4 pages, which unfortunately leaves several things unclear:  
  
I had difficulties in understanding how the questions and code examples were selected. Section 2.1 is very brief about this. Were the 75 questions selected randomly or in a systematic way? It appears that the selection of code examples was done systematically but this is only hinted at in Section 3 ("comparative quality analysis among the two code examples of the same question").  
  
It was not clear how the quality model was built. Section 2.3 suggest that 50 out of 110 code examples were used (again: how selected? why 50?) combined with ten-fold cross-validation. When using ten-fold cross-validation, ten models are built and evaluated. This is where it gets confusing: What happened with the results of the ten-fold cross validation? Based on what data was the model in Equation (2) built, one fold vs. the 50 code examples vs. all code examples?  
  
Why was the model in Equation (2) built without intercept? Not choosing an intercept can significantly change the value of a coefficients of a model. Furthermore, the coefficients of logistic regression models can only be directly compared if they have the same range (this is not the case as RV is not between 0 and 1).  
  
In Section 3, what model was used to create Table 1? If it is the model from Section 2.3 there appear to be some overlap between training and testing data.  
  
  
----------------------- REVIEW 2 ---------------------  
PAPER: 153  
TITLE: Reusability Analysis of Crowdsource Code Examples  
AUTHORS: Mohammad Masudur Rahman, Chanchal Roy and Iman Keivanloo  
  
  
----------- REVIEW -----------  
Summary:  
  
The authors are attempting to construct a metrics-based model for evaluating the quality of a code example provided in StackOverflow that can mimic the subjective judement of the community, and consequently assist in identifying useful examples for a problem for reuse.   
  
Review:  
  
The authors did themselves a disservice in titling their paper "Reusability Analysis" as they do not at all study the reusuability of the examples in question, but are instead interested in identifying the quality of the examples. This confused me, and made me wonder if the goal of the paper had changed during the writing. That said, it (quality analysis) is an interesting topic, and I think it is a worthwhile endeavour to try and determine the qualities that make examples useful to the end user. I think the authors attempt to explore some of these metics is worthwhile, and I would like to see more of that.   
  
I think the dimensions you're considering in your methodology are largely interesting, and I would love to see what other dimensions you come up with. It's a shame you do not report the scores on these dimensions, as I think it actually would be the best part of this paper. I was a little confused as the start of Section 2.3 promises 5 code-related metrics, yet you seem to only provide 4 metrics, until one realizes that "Code Soundness" includes 2. It does concern me that you are not self-critical of your metrics though: from your model, you suggest that AR has almost no influence on the outcome, yet you do not seem to wonder why that is? In the actual results however, RV (Rules Violation) is actually a negative addition to your model, yet there does not seem to be any discussion on this. You also seem, perhaps simply from lack of discussion otherwise, to think that your model is complete, and you do not need to discover other dimensions. This is most evident when talking abo  
ut why you do not have 100% success.   
  
I think your attempts to model the quality of examples are too premature, based on what you have presented. I think there is more value in discussing these dimensions, their correlation (or lack there of) and what that means about the nature of these examples, and especially what other factors could contribute to quality?  
  
  
----------------------- REVIEW 3 ---------------------  
PAPER: 153  
TITLE: Reusability Analysis of Crowdsource Code Examples  
AUTHORS: Mohammad Masudur Rahman, Chanchal Roy and Iman Keivanloo  
  
  
----------- REVIEW -----------  
SUMMARY:  
  
This paper investigates the quality of code examples from StackOverflow by evaluating them with current code metrics. It considers both the highly rated answers, promoted code example, and the negatively rated answers, discouraged code examples, to determine if the ratings are indicative of quality.   
  
The authors outline two research questions to relate the code quality and answer rating:  
  
1. Is the code level quality of a discouraged code example worse than that of a promoted code example?  
2. Why doesn’t the metric-based classification of the code examples completely agree with vote score-based classification on StackOverflow?  
  
The evaluation metrics are: readability, strength, weakness, coding rule violation, and one associated metric, like author’s expertise. They also create a model to determine the quality of the code examples with the metrics. They use their code model to determine if high quality code examples are actually correlated to high ratings by the community.   
  
The authors used the most recent StackOverflow data dump and picked 75 questions with more than 10 answers containing code examples from the dump to investigate. The code examples were manually analyzed to make them able to compile as well as filtered out the cases that were too generic so that they had 55 questions with applicable code examples. Additionally, they used the highly promoted and highly discouraged examples, which was determined by the vote score per day. To determine the weights of the metrics in the model, the authors preform logistic regression with 10-fold cross validation to create the model.   
  
The results indicated that 74.54% of the examples had agreement between the model and ratings of the StackOverflow community. So, the authors state that for their study there is a difference between code quality and it is worse for discouraged code in approximately 75% of cases. For the second research question, they considered the 28 code examples that the model quality disagreed with the community rating. They determine there models suffers from cases with lack of comments (used in their metrics), the use of relative quality (forcing promoted and discouraged examples), cases of trivial examples, and cases where code examples are not vital for the answer.   
  
COMMENTS:  
  
The novelty of the paper is in quantifying the quality of the examples from community-supplied answers from StackOverflow. The quality is determined by using code metrics and they investigate the connection between high quality code and highly voted answers.   
  
The research questions and descriptions of the metrics are clearly outlined. The vote-based differentiation makes logical sense in determining promoted or discouraged examples. Additionally, the code metrics are well motivated with related work and give clear indications of any alterations from pervious work.   
  
The use of logistic regression to build the quality model strengthens its applicability since the authors have an accepted basis for the weighting of each metrics. The high-level results of the logistic regression also seem valid since the intuitive relationships are shown in the model (e.g. readability is weighted more than author rank). When describing the usage of logistic regression, the threshold that was used should be given since it would impact the model by reducing either false positives or false negatives.  
  
For the evaluation, the combined metrics that the authors use and two other combinations suggest equal agreement percentage. The combinations all have readability and code soundness (strength and weakness). The rule violation seems to degrade the results and author rank does not change it, except when author rank and rule violation are added together. An explanation of these results would strengthen the evaluation since it primarily focuses just on the 74.54% of agreement.   
  
In terms of the second research question, the manual inspection explains the limitations and weaknesses of the model. However, it would benefit from displaying the occurrence of each “limitation” or analysis on the frequency of these cases. As currently stated, the second research question is weaker since it is directly implied by the results of the first question. A more quantitative question and analysis would strengthen evaluation of the model.   
  
Minor Comments:  
  
In section 3, the last sentence of the first paragraph says “in the next section…” when it should say paragraph since the next section is the conclusion.  
  
In section 3, the second paragraph says, “…about 75% times” but it should be ”…about 75% of the time.” Additionally, this whole sentence can be restructured to something like “Our preliminary results indicate that the code quality of discouraged code examples is worse than promoted code examples in about 75% of the examples.”